Ch-17

1] Inserting n elements using.

a] Aggregate Method :-

- the table doubles in size when it runs out of space

- so, if the original size is 1, after insertion it doubled to size 2, after 2 more inserting, it doubles to size 4, etc.

- In general, after k doublings, the size is 2^k.

Pseudo code :-

```
Initialize table with capacity = 1

For i = 1 to n
    if table is Full
        new table = create newtable with
                    size, 2 * count size
        copy elements from old table to
        new table
        table = new table

    insert element at i into table
```

Let $k = \log(n+1) - 1$

Total cost $= O(n) * k$
$= O(n \log n)$

Amortized cost per insertion $= O(\log)$

Runtime per insertion is $O(\log n)$

Total time is $O(n) * \log(n+1)$

b] Accounting Method

charge 2 units For each insertion.

when the table doubles in Size, From m to 2m, credit m units.

The credit exactly pay For the copy cost $O + O(m)$

Total credit is $m + 2m + 4m + \ldots n/2 * m$ $O(n)$

Pseudo code

initialize table with capacity = ✓

For i = 1 to n
    if table is Full!
    new table : create Newtable with
    size 2 * curmt size

    copy elements from old table to
    new table
    table : newtable

insert element + i into table

initialize charges = 0
initicllize credits = 0

For i = 1 to n:
    charges t = 2

if table doubled in size from m
to 2 m
    credits + : m

Total charges = 2 * n = o(n)

Total credits = m + 2m ... n/2^k m = o(n)

Amortized cost per insertion = Total/n
= o(n)/n
= o(1)

Runtime per insertion = O(1)

Total time = O(n)